Man and Mainframe at Dartmouth: Liberal Possibilities in Centralized Computing

Shane Shen Lin
Seattle, WA

Bachelor of Arts, University of Texas at Austin, 2007

A Thesis presented to the Graduate Faculty
of the University of Virginia in Candidacy for the Degree of
Master of Arts

Department of History

University of Virginia
May, 2014

Introduction

The woman with the wild blonde hair sprints through the marble corridor. The sledgehammer in her hands is almost comically outsized against her lithe and compact frame. She is dressed as an athlete–white tank top, fluorescent orange running shorts, sweat bands on both wrists of mismatched colors–but in place of a number, her shirt is adorned with a stylized personal computer and mouse. Although she is pursued by a phalanx of armed and armored guards, she seems neither worried nor weary. There can be no mistaking her for the anonymous gray crowd that she runs toward. They sit unmoving, even unblinking, as they watch the enormous screen in front of them dominated by a bespectacled guise. Imperiously, the man intones: “We are one people, with one will, one resolve, one cause.”

The woman objects. She spins to build momentum and, with a cry, looses the hammer at the screen as the guards reach her. It strikes just as the projected figure proclaims: “We shall prevail!” An explosion flares, finally wresting the somnambulant masses out of their stupor. An announcer unsubtly enunciates: “On January 24^{th}, Apple Computer will introduce Macintosh. And you'll see why 1984 won't be like '1984'.”

Aired during Super Bowl XVIII, Apple's “1984” ad was an immediate and stunning success. It departed from traditional advertising form by not showing the product promoted nor asserting any claims about its superior qualities. Instead, it elevated Apple's new computer as a strike against tyranny. In case there was any doubt about the identity of Big Brother, Apple co-founder Steve Jobs made the message explicit two days later at the annual Apple shareholders meeting where he unveiled the Macintosh computer. Presenting the future of personal computing as a Manichean struggle between Apple and IBM, Jobs accused his adversary of threatening the “future freedom” of computing. He asked the crowd: “Will Big Blue dominated the entire computer industry? The entire information age? Was George Orwell right?” To each question, the crowd shouted back “No!”

With its abstract message of individuality and freedom in place of any specific product claims, "1984" has far outlived the machine it was made to promote (Apple even resurrected it twenty years later in 2004, cheekily giving the runner a pair of its trademark white iPod headphones). Swift and nimble in spite of her massive payload, the runner became the most potent symbol of the distributed, individual, and personal revolution that swept the computing world over the next decade. With its legion of faceless drones and overbearing Orwellian authority, the ad also helped to cement a contrasting view of mainframe computing as monolithically conformist, strictly hierarchical, and corporate, the very enemy of discordant diversity.

This binary has proved remarkably resilient. The dim view of the mainframe echoed fears of earlier decades, of machines become too powerful to the despair of their creators – of HAL 9000 and Colossus and the Strangelovean Doomsday Device. With the dramatic downfall of mainframes and the corresponding ascension of personal computers, "1984" became prophecy, further reinforcing its divergent diptych. Moving forward, it has served as a useful foil for the new heroes of the PC revolution: Cambridge hackers, Menlo Park counterculturalists, and ascendant garage enterprises from Los Altos to Albuquerque. The ideas that they championed became seen not only as PC philosophies that sharply diverged from those of the mainframe, but as inherently and exclusively PC ones.

Computing at Dartmouth College, from its establishment in the late 1950s through the height of its mainframe time-sharing system in the 1970s, serves as a compelling counter to this strict dichotomy of PC and mainframe. The BASIC computer language was originally developed for use on Dartmouth mainframes before it became the *lingua franca* of the landmark 1977 personal computer "Trinity"; in fact, its later use by PCs was a direct consequence of the philosophical objectives of its creators. Precedents to many of the ideological shifts attributed to the later PC revolution–open access to computing resources, democratization of information, popularization of computer education, Free software–can all be found on the Hanover campus a decade before the rise of the personal computer. In challenging 1984's vision of mainframe computing, these convergences help remind us that although hardware architecture helps shape the contours of its use, it is not destiny. Within the allowances of technical possibility, the individual choices and institutional culture of its designers and users are important determinants to consider.

Early Years: 1955 to 1959

From its inception, computing at Dartmouth was driven largely by two professors of mathematics, John G. Kemeny and Thomas E. Kurtz. Kemeny had immigrated with his Jewish family to the United States as a teenager, leaving his native Hungary just prior to its entry into World War II. Near the end of the war, he was drafted into the US Army, which happily recognized his mathematical talents and assigned him to Los Alamos. There, Kemeny's work with the Manhattan Project's team of human computers and his acquaintance with fellow Hungarian émigré and computer pioneer John von Neumann would guide his later enthusiasm for digital computation. After the war, Kemeny briefly worked as Albert Einstein's assistant at Princeton, completed his doctoral thesis under Alonzo Church (who earlier helped establish the theoretical foundations of computer science with his contributions to the Church-Turing thesis) in under a year, and was appointed assistant professor of logic in the Philosophy Department at Princeton in 1951. Two years later, on the recommendation of Einstein and von Neumann, Dartmouth College recruited Kemeny by offering him a full professorship and near-total autonomy to completely rebuild its aging and rapidly shrinking mathematics program as departmental chairman. They were unaware when they made the offer that he was only 27 years old.[1]

---

1 Kemeny unpublished autobiography, Section 6, Box 20, Manuscript MS-988 Papers of John G. Kemeny, 1942 - 1992, Dartmouth College Rauner Special Collections Library (hereafter cited as Kemeny Papers). Chapters “Member of the Faculty” 6-7, “Dartmouth” 1-3
Kemeny, John G. *Man and the Computer* (New York: Scribner, 1972), 1-10.

Two recent Princeton Ph.Ds, John McCarthy and Thomas Kurtz, were in the first wave of mathematics faculty that Kemeny hired and showed special interest in computation. McCarthy was drawn to the idea of artificial intelligence (a term that he helped coin) and, on arriving at Dartmouth, immediately organized the Dartmouth Summer Research Project on Artificial Intelligence, a seminal event in establishing AI as a formal field of research. Kurtz had earlier worked with the National Bureau of Standards's early, custom-built Standard Western Automatic Computer (SWAC) during a summer stay at the University of California at Los Angeles. Concurrent with Kurtz's hiring in 1956, the Massachusetts Institute of Technology and IBM Corporation established the New England Computation Center, a shared computing resource for regional universities; even prior to arriving at Dartmouth, Kurtz was diverted to MIT to learn how to use the IBM-704 computer at the heart of the Center, a similar type of machine as Kemeny had been exposed to a few years earlier as a consultant for the RAND Corporation.[2]

Over the next three years, Dartmouth became the New England Computation Center's most active remote participant despite the onerous burden of geography; Dartmouth users of the Center's IBM 704 computer were forced to send code on IBM punch cards through the mail or else make the four hour drive from Hanover to Cambridge. During this time, Dartmouth's computer use typically consisted of mathematics research ("calculating the interesting statistical properties of finite Markov Chains") in part because of a technical barrier to entry. The IBM-704 operated through the complicated Symbolic Assembly Program (SAP), a low-level assembly language system that required intimate technical knowledge of the hardware

---

2 McCarthy would shortly leave Dartmouth for M.I.T., in part because of Dartmouth's lack of a computer. There, he would help found the influential A.I. Lab, invent garbage-collection, create the LISP programing language, and help develop time-sharing before moving to a more permanent home at Stanford.
Shasha, Dennis Elliott, and Cathy A Lazere. *Out of Their Minds : the Lives and Discoveries of 15 Great Computer Scientists* (New York: Copernicus, 1995), 22-26.
McCarthy, J. et al. "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence", August 31, 1955.
Szczepaniak, John. "Basic history of BASIC." Game Developer Magazine, p. 37-38
Kemeny unpublished autobiography, Section 6, Box 20, Kemeny Papers. Chapter "Dartmouth", 6-7

architecture of the computer to perform even simple operations. Contrary to this hurdle, Kemeny and Kurtz shared a common vision of computing as more than an insular field and of computers as more than expert tools; rather, they felt that programing should be a generalized skill useful for many different disciplines and functions, including those outside of the sciences. Although the inconvenience of distance was impractical to overcome, the difficulty of coding for the 704 was more open to remedy.[3]

Within a few months, Kemeny and Kurtz devised a stopgap solution: DARSIMCO, the Dartmouth Simplified Code. It is to this project that they later traced the roots of "the genealogy of BASIC". DARSIMCO was an extended set of instructions in the same format as SAP, but which obscured the complex underlying hardware mechanics. These were intended to be used in sets of reusable templates corresponding to simple arithmetic operations, so that novice users totally ignorant of the meanings of individual instructions could still code. The technical ideas behind DARSIMCO weren't very revolutionary and it didn't catch on; both it and SAP were quickly superseded by FORTRAN, the first popular general-purpose high-level programming language developed by IBM's John Backus. FORTRAN simplified coding along similar lines as DARSIMCO, but much more comprehensively by entirely breaking out of SAP's mechanical conventions (Backus had also worked with some of the 704's designers to better support FORTRAN's functionality in hardware). It allowed programmers to use familiar English keywords and mathematical syntax to compose code and then transform that code to hardware-specific low-level language for execution. DARSIMCO's development, although unsuccessful in the face of FORTRAN's much more extensive solution, nonetheless demonstrated Kemeny and Kurtz's dedication to expanding the scope of computer users; a DARSIMCO instruction manual from 1956 freely offered Kurtz's time in assisting "any" of a "wide variety of Dartmouth

3 "Computing at Dartmouth: Past – Present – Future", Box 10, Manuscript MS-1144 Papers of Thomas E. Kurtz, 1950-2008. Dartmouth College Rauner Special Collections Library (hereafter cited as Kurtz Papers).
Kemeny unpublished autobiography, Section 6, Box 20, Kemeny Papers. Chapter "Computers", 1-2
Kurtz file memo: "Newsletters from Royal McBee and POOL", Box 3, Kurtz Papers.

faculty members" in the use of it. The fact that they were ultimately dissatisfied with FORTRAN in spite of its widespread popularity and pursued new approaches to better realize their goals elucidates the subtle ways that these goals differed from those of IBM and FORTRAN.[4]

Dartmouth College purchased its first computer in 1959, coincident with the construction of the Bradley Center for Mathematics. Kurtz later suggested that Kemeny had persuaded university administrators to divert a portion of the budget for the building's furnishings to pay for the machine. Kemeny, Kurtz, and their wives drove to Boston to negotiate the purchase of a Royal McBee LGP-30. The desk-sized LGP-30 was physically a stark contrast to the IBM-704, which could fill a large room, and the difference in cost and performance was just as striking. Though prices varied by customer and configuration, the LGP-30 sold for roughly two orders of magnitude less than the IBM. Writing on it fondly decades later, Kemeny described the machine as "a toy" and even sympathetic contemporary accounts acknowledged that it was "slow".[5]

Yet this relative impotence was a strength of a kind. Computationally complex code could still be sent to MIT. Beyond ameliorating the difficulties of distance, the LGP-30 was intended to address a specific shortcoming of using a remote mainframe: Dartmouth has historically strongly emphasized liberal undergraduate education (the mathematics department only began to accept Ph.D students around this time) and a faraway IBM-704 was not very useful for familiarizing college students with computing. In the context of this issue, the decision to purchase a LGP-30 was entirely in keeping with Kemeny and Kurtz's goal of making

4 Kemeny, John G., and Thomas E. Kurtz. *Back to Basic: the History, Corruption, and Future of the Language* (Reading, Mass.: Addison-Wesley Pub. Co., 1985), 6-7.
"DARSIMCO: Dartmouth Simplified Coding Approach For the IBM 704, October 16, 1956", box 10, Kurtz Papers.
IBM Archives: John Backus, IBM, last modified May 10, 2013, http://www-03.ibm.com/ibm/history/exhibits/builders/builders_backus3.html

5 Kemeny recounted that the four of them took a single car to Boston and brought the computer back with them. Kurtz recalled, "I am certain we did not carry it back to Hanover ourselves." Given the sizable bulk of the LGP-30 and its weight of over 700 pounds, it's more likely that Kemeny is embellishing for the sake of a good story.
Kemeny unpublished autobiography, Section 6, Box 20, Kemeny Papers. Chapter "Computers", 1-2
Kurtz file memo: "Newsletters from Royal McBee and POOL", Box 3, Kurtz Papers.
Levy, Steven. *Hackers: Heroes of the Computer Revolution* (Garden City, N.Y.: Anchor Press/Doubleday, 1984), 4-5.

computing accessible to a larger population outside of the arcane worlds of electrical engineering and theoretical mathematics. The new building wasn't yet ready, so the computer was installed in a 120 square-foot room in the basement of College Hall (thankfully, the LGP-30 lacked the exceptional power or cooling requirements of more powerful computers). This closet became the material component of the Dartmouth Computation Center, of which Kurtz was named the director. These humble origins quickly yielded impressive results.[6]

Permission to use the computer was remarkably free. An early copy of the Computation Center procedure manual is clear: "The computing machine and its related equipment are available to anyone at Dartmouth College who is interested." The rest of the short manual outlines technical maintenance and housekeeping tasks, covers error reporting and user logging, and prohibits the "borrowing" of office supplies; the only formal restriction of use is the requirement for students to carefully read the document itself. After its first year of operation, Kurtz reported that the machine had been demonstrated to about a hundred students from the college and the local high school and a further fifty students achieved a very basic degree of programing ability through coursework via the mathematics department and the Thayer School of Engineering. Above these groups and in spite of the lack of a formal course at the college dedicated to computation, "about fifteen or twenty students" had become "expert programmers through prolonged, entirely extracurricular contact with the Computer".[7]

In some respects, the LGP-30 was both a step forward and one back. Although the machine was now physically accessible to Dartmouth users, its use was again dependent on a hard-to-code assembly language, much like in the early days of the IBM-704. Much of the most productive work by the new class of student

---

6 Kemeny, John G. *Man and the Computer* (New York: Scribner, 1972), 8-9.
"Computing at Dartmouth: Past – Present – Future", Box 10, Kurtz Papers.

7 LGP-30 Procedure Manual, 1960, Box 3, Kurtz Papers.
Kurtz, "Report on the Activities of the Computation Center, 1959-1960", Box 10, Kurtz Papers.

experts was dedicated to remedying this. One such student, Robert Hargraves Jr., wrote DART over the summer of 1959, virtually as soon as the machine was installed. DART shared some of the same goals as Kemeny and Kurtz's DARSIMCO on the IBM-704; it was intended to allow "inexperienced programmers" in a variety of fields to solve numerical and statistical problems by reducing the complexity of low-level coding. But going beyond the simple convenience functionality of DARSIMCO, DART implemented its own grammar, seemingly influenced by some of FORTRAN's keywords, to streamline algebraic calculation and control program flow. There was even limited support for subroutining. Compared to FORTRAN, its grammar was limited and crude and it's doubtful that its output was well optimized; however, it's also clear that DART has much more in common with the IBM language, in function and design, than Kemeny and Kurtz's facile early effort.[8]

Despite its clunky language, DART was an impressive feat for an undergraduate student without formal training in programing. In reporting it to the College, Kurtz was effusive with praise: DART was "a real and important contribution to the scientific community" and that "such activities obviously should be encouraged". The next step was even ambitious. For several years, an international standard computing language had been in the works in order to normalize both the theoretical discussion of algorithms and to smooth out the divergent details of hardware in practical implementation. This effort was finalized in 1960 as ALGOL 60. Unlike the artless simplicity of DART, ALGOL was designed by accomplished experts, including IBM's John Backus and John McCarthy (then at MIT), to serve the diverse and complex needs of academic research. A group of four undergraduate students, including Hargraves, quickly proposed implementing the demanding new language for the underpowered LGP-30 with the intention that it could become a new standard language for Dartmouth mathematics students. Years later, Kemeny reflected on their enthusiasm:

8 Kurtz file memo, LGP-30 Projects, box 3, Kurtz Papers.
Hargraves, "DART I", box 10, Kurtz Papers.

"Anyone could have told them that this was ridiculous, but Tom [Kurtz] did not tell them that. So, not knowing that their goal was impossible, they succeeded!"[9]

The students faced a number of challenges, ranging from the machine's impoverished memory and lethargic pace (even the much simpler DART had severe restrictions on the number of operations and statements allowed in any given program) to the fact that the ALGOL grammar required mathematical symbols not physically found on the LGP-30's keyboard. Although accommodations had to be made to overcome or circumvent these issues, the resulting language contained the bulk of ALGOL's features; in recognition of the compromises necessary in compressing ALGOL 60 to the LGP-30, it was dubbed ALGOL-30. ALGOL-30's main flaw was that memory limitations required users to undergo a "two stage" process, where the compiled program had to be written to paper tape before being loaded back into the computer to be run, an inconvenience that limited its potential to teach programming to undergraduates. To overcome this, the student programmers ensured that the design of ALGOL-30's successor, the Self-Contained ALGOL Processor (SCALP), was sufficiently streamlined to allow "load-and-go" operation without any intermediate step.[10]

SCALP was an impressive achievement. Although Kurtz supervised both ALGOL-30 and SCALP, he was quick to emphasize that Dartmouth students accomplished all the planning, coding, and debugging for both projects. Concurrent with their development, several other commercial languages for the LGP-30 arose

---

9 Hargraves would go on to teach at Dartmouth and become Associate Director of the Kiewit Computation Center.
Kurtz, "Report on the Activities of the Computation Center, 1959-1960", Box 10, Kurtz Papers.
Nauer, Report on the Algorithmic Language ALGOL 60, Numerische Mathematil. 2,105 136(1960)
Kemeny unpublished autobiography, Section 6, Box 20, Kemeny Papers. Chapter "Computers", 1-2
Wexelblat, Richard L. *History of Programming Languages* (New York: Academic Press, 1981), 516-517.

10 Kurtz file memo: "LGP-30 Projects", box 3, Kurtz Papers.
Kurtz to University Computing Center Directors, "ALGOL Compiler Progress Report", Dec. 8, 1960, box 10, Kurtz Papers.
"Computing at Dartmouth: Past – Present – Future", box 10, Kurtz Papers.
"Programming in the ALGOL-30 System for the LGP-30", box 3, Kurtz Papers.
Kurtz, "ALGOL for the LGP-30: A Comparison", Feb. 15 1962, box 3, Kurtz Papers.
Wexelblat, Richard L. *History of Programming Languages* (New York: Academic Press, 1981), 516-517.

to fill needs similar to those that SCALP addressed, some of which also emulated ALGOL. Kurtz was able to proudly claim that even pitted against "expert professional programmers", his undergraduates were able to achieve faster speeds and more robust features. SCALP was quickly integrated into the mathematics curriculum at Dartmouth and, over the next few years, a sizable number of math students learned to code through it.[11]

SCALP was a powerful tool to help students solve number theory proofs and other theoretical math problems. But mathematics and engineering students constituted only around 25% of Dartmouth's student population; the complexities of ALGOL's abstract math features did not perfectly suit Kemeny and Kurtz's goals of an even broader computing constituency. From the start, interdisciplinarity was a key component of this. In his first report on the Computation Center in 1960, Kurtz emphasized that many different departments were taking advantage of the new LGP-30 with the help of undergraduate assistants. Beyond mathematics and engineering, these included mineral studies for the geology department, economics modeling of wages and consumption, medical research on heart disease, and even some work for admissions. Diversity was an important objective for Kemeny and Kurtz, and not just to curry board, cross-departmental support for their computing project at Dartmouth. Kurtz later wrote of how they had fretted over how best to teach computing to the nonscience students who would go on to become decision-makers of business and government, for "how can sensible decisions about computing and its use be made by persons essentially ignorant of it?" Kemeny produced an essay in 1963 with much the same sentiment, adding that including computer training as an integral part of a complete liberal arts education will save those future executives and

11 "A Manual for SCALP: being a Self Contained ALgol Processor for the General Precision LGP-30", box 10, Kurtz Papers.
"Computing at Dartmouth: Past – Present – Future", box 10, Kurtz Papers.
Kurtz to Gafney, May 24, 1961, box 10, Kurtz Papers.
Wexelblat, Richard L. *History of Programming Languages* (New York: Academic Press, 1981), 516-517.

policy-makers from sharing "with the older generation the basic fears of man-machine relationship." For computing to be truly influential, it needed to be universal.[12]

Beyond expanding computing within Dartmouth, the Computation Center also worked to increase exposure to computers at the secondary education level. In the summer of 1961, the LGP-30 was used as part of an eight-week NSF program to train high school teachers from across the country in computing. Dartmouth also accommodated particularly gifted local high school students. Kemeny tasked one of these students, Sidney Marshall, to implement a simple language suitable for teaching the basics of programming to college freshmen of all fields over the course of three one-hour lectures as well as for simple research problems. Kemeny helped in its design and named it DOPE: the Dartmouth Oversimplified Programing Experiment. DOPE's twin goals would prove difficult to reconcile: the rigidity of the language and its limited feature-set made it easy to teach to beginners, but not very useful beyond. Although many of its design features presaged those later found in BASIC, the lone LGP-30 computer that hosted it was already increasingly busy to the point of saturation by the time it was completed in 1962 and DOPE was never widely used.[13]

These early projects all demonstrated Dartmouth's dedication to expanding the use and understanding of computing, not just by making resources openly available, but by actively ensuring that those resources were as broadly used and useful as possible. And while projects like DART, ALGOL-30, SCALP, and DOPE are often cited as antecedents of BASIC, their development as student projects is often overlooked. In many cases, it was the students who took the initiative, taking advantage of the permissive environment provided by the faculty. Years before the height of the student protest movement, before Mario Savio lamented the lack of

---

12 Wexelblat, Richard L. *History of Programming Languages* (New York: Academic Press, 1981), 518.
Kemeny, "Computing Center at a Liberal Arts College", collected in Kemeny, *Random Essays on Mathematics, Education, and Computers* (Englewood Cliffs: Prentice-Hall, 1964), 157-163.

13 "Computing at Dartmouth: Past – Present – Future", box 10, Kurtz Papers.
Kemeny unpublished autobiography, Section 6, Box 20, Kemeny Papers. Chapter "Computers", 2
"Dartmouth Oversimplified Programming Experiment", May 1962, box 3, Kurtz Papers.

student volition by declaring that "at Cal you're little more than an IBM card", Dartmouth undergraduates were not only writing the code, but designing the systems on which it ran.[14]

Development of Time-Sharing: 1962-1964

By the end of 1961, it was already clear to Kemeny and Kurtz that further expanding access to computation at Dartmouth was exceeding the capabilities of their lightweight LGP-30. A typical solution to the problem of serving a large number of users with single machine was batch processing, wherein a large number of programs were scheduled and run together to reduce the burden of unnecessary human delays on the system. However, the pair felt that the waiting time this method required was unsuitable for the trial-and-error coding style of beginning students, particular those without a rigorous background in science and mathematics. Under batch processing, users would have to wait their turn for their code to run and, if they made mistakes, would have to wait again to schedule every correction. Depending on the schedule, a user might be scheduled for three runs in a week, requiring many weeks to completely debug a complex program even if the total compute time was very brief. John McCarthy suggested looking into new research at MIT on time-sharing, a new method of computing where the mainframe switched invisibly between multiple simultaneous users working independently on different access terminals. The key advantage of the system was essentially instantaneous turn-around times, which would, in Kurtz's memorable phrasing, allow the "luxury of 'learning by making mistakes'". Students could then teach themselves through a high degree of interaction rather than onerous preparation reliant on theory and formal documentation.[15]

14 "Angry Words from Mario Savio, Spokesman for California's Students Now Facing Trial", Life Magazine, Feb 25, 1965. Quoted in: Turner, Fred. *From Counterculture to Cyberculture: Stewart Brand, the Whole Earth Network, and the Rise of Digital Utopianism* (Chicago: University of Chicago Press, 2006), 12.

15 Kemeny unpublished autobiography, Section 6, Box 20, Kemeny Papers. Chapter "Computers", 2-4
"Proposed Computation Center at Dartmouth College based on Computing Equipment by General Electric", box 1, Kurtz Papers.

In 1962, Kemeny and Kurtz proposed Dartmouth Time-Sharing, a system which explicitly prioritized the goal of universal undergraduate access over even Dartmouth faculty research. With the LGP-30 stretched to its limits, they needed to evaluate and purchase new hardware. As with prior computation projects at Dartmouth, students played a key role in the development of this system, even at the most preliminary stages. In the summer of 1962, shortly after McCarthy's suggestion, Kurtz and student Anthony Knapp (one of the developers of SCALP), visited General Electric's computer division in Phoenix, Arizona to examine their computer offerings. With the hope of enticing General Electric as a research partner and obtaining for Dartmouth a computer as a donation, Knapp sketched a preliminary outline for a time-sharing system operating on GE hardware named the Master Executive Supervisory System. Presented with a proposal titled MESS (continuing the tradition of tongue-in-cheek Dartmouth acronyms like SCALP and DOPE), GE unsurprisingly declined to any interest in a joint venture. However, Kemeny and Kurtz would later cite Knapp's proposal as instrumental to convincing them of a general-purpose time-sharing system was possible with GE's modest hardware.[16]

Following a brief review of potential vendors, General Electric's GE-225 and Datanet-30 computers were ultimately selected as the best choice (despite somewhat acrimonious back-door politicking by IBM). The GE-225 would serve as the main processing unit while the Datanet-30 would operate as its time-sharing master; users would communicate with the Datanet-30 by teletype terminals and the Datanet-30 would then schedule their operations on the GE-225. Dartmouth negotiated a slight discount beyond the standard educational markdown and signed the sales contract, totaling around $441,000 with fifteen terminals, in April

16 "Kiewit Computation Center Fiscal 1967-68 Annual Report to the Trustees of Dartmouth College", box 6, Kurtz Papers.
Wexelblat, Richard L. *History of Programming Languages* (New York: Academic Press, 1981), 532-533.
Kurtz, file memo "1962"
J G Kemeny and T E Kurtz, "Dartmouth Time-Sharing: Development of the system by a team of faculty and undergraduates is described", *Science*, November 1968, p. 223-8.

of 1963. To pay this expense (about an order of magnitude greater than had been spent on the LGP-30), the computers were rented for a year with the option to buy contingent on the approval of two NSF grants. While the NSF reviewers were supportive of Time-Sharing's stated goal of expanding access to computing to more than 70% of Dartmouth students, they were more skeptical of Dartmouth's proposal to call on that same student body to develop the system over a very short period. Kemeny later recalled that one exasperated reviewer complained that they were "not even graduate students but undergraduates!" Fortunately, the proposal was approved.[17]

The students didn't disappoint. While Kemeny and Kurtz provided the general design for the system, the implementation fell to them. The GE-225 arrived at the end of Feburary 1964 and, by Kurtz's account, the students (including Sidney Marshall, the developer of DOPE and now a Dartmouth undergraduate) immediately "swarmed" over it. They overcame significant obstacles over many all-night coding sessions; some worked more than 50 hours in a week while still responsible for classwork. Neither the GE-225 nor the Datanet-30 had been designed for time-sharing operation and many functions, such as coordinating data storage between the two machines, that would typically be handled by hardware had to be controlled with software. The first time-shared program successfully ran in May and by the beginning of the Fall semester Time-Sharing was ready for operation.[18]

---

17 Kemeny was sufficiently incensed by IBM's lobbying to accuse an aggressive attack of the GE decision in favor of IBM's 7040 computer of being "grossly misleading", "simply ludicrous", and a conservative hedge against real innovation. Kemeny to Provost Masland, et al. "Response to memo by Miles Hayes", May 27, 1963.
Kurtz, file memo "1962", box 3, Kurtz Papers.
Wexelblat, Richard L. *History of Programming Languages* (New York: Academic Press, 1981), 532-533.
Kemeny unpublished autobiography, Section 6, Box 20, Kemeny Papers. Chapter "Computers", 3-4.
"Outline of a Plan for the Central Computation Center for Dartmouth College", box 1, Kurtz Papers.

18 In keeping with the precedent set by the ad-hoc installation of the LGP-30, the GE-225 necessitated "emergency" alterations to the "horrible old basement" of College Hall in order to accommodate it.
Kemeny unpublished autobiography, Section 6, Box 20, Kemeny Papers. Chapter "Computers", 5-8
Kurtz, file memo "1962", box 3, Kurtz Papers.
J G Kemeny and T E Kurtz, "Dartmouth Time-Sharing: Development of the system by a team of faculty and undergraduates is described", *Science*, November 1968, p. 223-8.

While the students worked on the Time-Sharing operating system, Kemeny and Kurtz turned their attention back to their perennial problem: a language easy enough for anyone to learn, but suitable even for most expert users. The language would also need to be a general-purpose language, so that any program can be written, take full advantage of the interactivity offered by time-sharing, be fast and clear in its messaging, and completely insulate the user from the specific details of hardware. DART, ALGOL-30, SCALP, and DOPE had all been unsuccessful at achieving all of these goals. Kurtz initially suggested that a subset of FORTRAN or ALGOL might be suitable, especially so that students could learn a language used outside of Dartmouth. Kemeny convinced him that a new language was necessary; there were just too many "ugly features" that had to be eliminated that any resulting derivation would be incompatible with its goal. A new language could also be designed from the ground-up to take advantage of Time-Sharing's interactivity and instantaneous response time. Completed over the course of a few months in early 1962, they dubbed the result BASIC: Beginner's All-purpose Symbolic Instruction Code. Among its features were a line number system to allow easy on-the-run modification of code, helpful and expressive error messaging so that users could make use of time-sharing's immediate feedback for quick debugging, and, eventually, an input mechanism to query users for information while the computer waited.[19]

The importance of BASIC, the component of Time-Sharing most completely attributable to Kemeny and Kurtz, has become outsized due to its later success and the degree of its necessity has often been overstated. In a prioritized list tasks from 1963, development of the language was listed last, behind the creation of the time-sharing system and work on the undergraduate computing curriculum. The two NSF grant proposals focus much more on the details of the hardware and the operating system that connected its various

19 J G Kemeny and T E Kurtz, "Dartmouth Time-Sharing: Development of the system by a team of faculty and undergraduates is described", *Science*, November 1968, p. 223-8.
Wexelblat, Richard L. *History of Programming Languages* (New York: Academic Press, 1981), 518-526.
Kemeny and Kurtz, *Back to BASIC*, 9-13.

components than on BASIC. In a 1963 memo, Kurtz even insisted that "the pedagogical distance between … ALGOL and an optimally designed simplified language is very small." And although BASIC was the flagship language supported for Time-Sharing, the system itself was always intended to (and very quickly did) support other languages. This is not to question whether BASIC achieved its goals; its widespread success helps to validate that. However, the degree to which BASIC represented a substantial innovation in balancing simplicity and power and the degree to which other contemporary languages failed to match it in this criteria are questionable.[20]

With both major components complete, Dartmouth Time-Sharing was ready for use. In preparation, Kemeny and the university leadership enumerated a few core principles for its operation, notably: students should have free access; they should have complete privacy, including from the faculty; the system should be easy to use; and priority should be given to students' convenience rather than the computer's efficiency. These principles were informed by the critical importance that Kemeny and Kurtz placed on the role of computers in a liberal arts education such that a system like Dartmouth Time-Sharing should be considered equal in stature on a campus to its library. And so, like a library, computation should be both free in cost and in latitude of access for students. Kemeny would often fall back on the library analogy. Questioned on the wisdom of an open system, he quipped: "Would you charge a student for taking a book out of the library? Sometimes I am tempted to *pay* a student to take a book out…" Kemeny would later go as far as to publicly propose that colleges and universities without adequate computer facilities be barred from accreditation, lest their students graduate "a whole era out of date". The last principle was a direct answer to the contemporary criticism that time-sharing was less efficient than batch operation: Kemeny and Kurtz believed that such a calculus inappropriately placed

20 "Computer Project Meeting Notes: Monday, October 7, 1963", Kurtz Papers.
"Kurtz to Project, "Subject: Languages" May 20, 1963", Kurtz Papers.
"Kurtz, Time Sharing Project Memo No. 1, Nov. 6 1963", box 3, Kurtz Papers.

the function of the machine over the experience of its users. Ideally, mainframe should serve man rather than the other way around.[21]

Universal Access at Dartmouth and Beyond: 1964-1969

Time-sharing was immediately put to work during the 1964-65 academic year. At the time, 80% of Dartmouth students completed the freshmen mathematics courses, so these were the ideal environments with which to experiment with universal computer access. Following the standard set by DOPE, the 10-week course would begin with three introductory one-hour lectures covering the basic operations of the language. Thus armed, each student would be scheduled for a half-hour block every week for the remaining nine weeks to code and debug four programs coordinated with the topics covered in the courses–calculus for science students and finite mathematics for non-science ones. Following the course evaluations from the first year, they reduced the number of lectures to two but lengthened coding sessions to 45 minutes per week; the professors found that while students were able to grasp the fundamentals of BASIC programing even more quickly than expected, they had misgauged just how poorly the students could type. The thousands of programs that these freshmen produced every year were subsequently checked by testing programs run on time-sharing. By 1968, the fourth cycle of this course, 80% of Dartmouth students had completed the program.[22]

---

21 Kurtz noted that such free and open access at Dartmouth was specifically made possible by the absence of government research at the College as was common at larger research universities, which required more stringent accounting for computer time lest taxpayers' dollars become frivolously wasted. However, in the first annual report on the Kiewit Computation Center for 1967-1968, he mentions in passing that the Center did help perform research for the Department of Defense as part of Project Themis, aimed at expanding military research grants to a wider array of universities.
Kemeny and Kurtz, "A Proposal for a College Computation Center", c1963, Kurtz Papers.
Kemeny unpublished autobiography, Section 6, Box 20, Kemeny Papers. Chapter "Computers", 11.
Wexelblat, Richard L. *History of Programming Languages* (New York: Academic Press, 1981), 519-520.
Kemeny, "Use, Non-use, and Misuse of Computers by Colleges", Conference on Computers in the Undergraduate Cirricula, June 1971", box 14, Kemeny Papers.

22 J G Kemeny and T E Kurtz, "Dartmouth Time-Sharing: Development of the system by a team of faculty and undergraduates is described", *Science*, November 1968, p. 223-8.
Kemeny and Kurtz, *Back to BASIC*, 10.
Overheads, Fall Term Course, Nov 1964. Kurtz papers.

Kemeny and Kurtz also continued to take pains to build a broad base of support for computation across many departments. The inaugural advisory Committee on Computing consisted of faculty from the medical, engineering, and business schools and the mathematics, economics, and physics departments. The pair conducted a series of Time-Sharing training sessions for faculty starting in June of 1964, almost as soon as it was operational. Kemeny was a lively and persuasive lecturer. One attendant of that first session, art and architecture historian Hugh Morrison, who had included himself as among those "least likely to benefit from a computer", became a convert after he posed a question ("What was the day of the week of the St. Bartholomew's Day Massacre?") and then quickly received an answer that adjusted for the switch from the Julian to Gregorian calendars. Afterward, he reported: "The course profoundly affected the thinking of us all … [the computer] is a weapon of such power that all intelligent men and women everywhere should know the kind of things it can do … This machine of ours is a friend." By 1968, half of the academic departments at Dartmouth included "official" computer use in a total of around 100 courses, ranging from Accounting and Finance and Econometrics to Aquatic Ecology and a course in the Classics department on the works of Virgil; almost half the faculty and administrative staff of the College joined their students in learning to use the system.[23]

The success of the freshman math classes boldly demonstrated that universal computer education was practical. Dartmouth's work quickly caught General Electric's attention. By the end of 1964, they were running their own copy of Dartmouth Time-Sharing in Phoenix. In September of 1965, GE offered Dartmouth the

---

Kemeny and Kurtz, "The Dartmouth Time-Sharing Computing System developed under a grant from the Course Content Improvement Program, National Science Foundation, Final Report, June 1967", box 11, Kurtz Papers.

"The TEACH System", box 11, Kurtz Papers.

23 It turns out, the answer that the computer gave for the day of the St. Bartholomew's Day massacre on August 24, 1572 (Wednesday) doesn't seem to be correct. The correct answer under the Julian calendar is Sunday. Morrison quoted in O'Connell, George. "The Computer Revolution", Dartmouth Alumni Magazine, Nov 1964.
Kemeny and Kurtz, *Back to BASIC*, 10.
Zani, William M. Memo received June 3, 1964, Kurtz Papers.
"Kiewit Computation Center Fiscal 1967-68 Annual Report to the Trustees of Dartmouth College.", Kurtz Papers.

deal it had been after all along: a joint venture. In exchange for Dartmouth's assistance in developing a more robust time-sharing system, GE would effectively donate a multi-million dollar GE-635 computer to Dartmouth. Construction tycoon (and Dartmouth alum) Peter Kiewit put up the money for a proper building on Main St., adjacent to Dartmouth's iconic Baker library, to house the new computer. Students (including Sidney Marshall of DOPE) again formed the bulk of the development team. The partnership diverged amicably after 1967; Dartmouth had gained a powerful and expensive computer that would serve it for the next decade and GE a robust and profitable time-sharing operating system to market both its computers and new computing service. A two-page 1968 advertisement for the General Electric Time-Sharing Service shows Kemeny at a teletype printer with a roomful of Dartmouth students. It quotes Kemeny: "Any one of these Dartmouth sophomores... starting from scratch... can do all the calculations in one afternoon that took a year and a half at Los Alamos." Leveraging the power of time-sharing, an enterprising student could best the largest centralized scientific project in human history.[24]

To ensure that BASIC could better achieve its goals of universal accessibility, Kemeny and Kurtz had placed it in the public domain. Due in part to this decision, BASIC quickly gained wide acceptance beyond the success of time-sharing at Dartmouth and GE. Its growing popularity served to resolve Kurtz's fear that they would be teaching their students a language without any direct application outside of Dartmouth. Even IBM jumped on the bandwagon. Unfazed by Dartmouth's rejection of their hardware in 1963, they released a version of BASIC for their own time-sharing hardware, claiming in an advertisement: "Start learning it after

24 Kemeny and Kurtz later recalled that the joint project resulted, for GE, in the largest commercial time-sharing network in the country. The value of the GE-625/635 computer is reported as varying amounts, from $2.5 million to $4 million. In his reports, Kurtz conservatively estimates the GE-635 to be eight times more powerful than the GE-265 it replaced.
Kemeny and Kurtz, *Back to BASIC*, 21-24.
Kemeny unpublished autobiography, Section 6, Box 20, Kemeny Papers. Chapter "Computers", 8-9.
"College and G.E. Collaborate on Major Computing Facility.", Kurtz Papers.
Wexelblat, Richard L. *History of Programming Languages* (New York: Academic Press, 1981), 532-533.
"General Electric Information Services advertisement, Jan. 1, 1968.", Kurtz Papers.
"Kiewit Computation Center Fiscal 1967-68 Annual Report to the Trustees of Dartmouth College.", Kurtz Papers.

breakfast and you can share our computer before lunch." Other implementations of BASIC and Dartmouth Time-Sharing were spread by Dartmouth students after their matriculation. One former Dartmouth student recreated time-sharing on the GE-415 for his new employers at Fulton National Bank in 1967; reflecting either supreme optimism or more pedestrian oversight, his reference manual section on the "Limitations on BASIC" was a blank page.[25]

With broad institutional support and freed from penny-pinched equipment and impromptu basement rooms, computation at Dartmouth gained a new measure of prominence. To dedicate the new Kiewit Computation Center, Dartmouth organized a conference in December 1966, featuring luminaries like computing pioneer George Stibitz, MIT AI Lab co-founder Marvin Minsky, information theorist Richard Hamming, and MIT's Robert Fano, whose early work on time-sharing formed the basis for Dartmouth's system. Instead of discussing narrow technical matters, the conference, titled *The Future Impact of Computers*, focused on the influence of computers on education, society, and knowledge itself. The Dartmouth participants used the opportunity to recommit the Center to Dartmouth's previous principles of free and open access. Myron Tribus, dean of Dartmouth's Thayer Engineering School and longtime partisan for computation at the College, placed computer access along with the library and student health services as fundamental institutional commitments to all future Dartmouth undergraduates and predicted that investment in computing would open new intellectual frontiers in the same way that the steam engine spurred the development of physics. Donald Kreider, of the Dartmouth math department, recounted Dartmouth students' early achievements on the LGP-30 absent strict supervision and cited the possibility of "wasted time, extensive trial-and-error, and even a certain amount of mischief" on a large scale as the keys to unlocking true potential of computers. Kemeny wrapped up the conference with sweeping predictions for computing in the future

25 IBM CALL/360:BASIC advertisement, June 20, 1968, Kurtz Papers.
'BASIC' Language Reference Manual, Fulton National Bank, March 1967, Kurtz Papers.

world of 1990, developed along the trajectory that Dartmouth Time-Sharing had set. He foresaw a connected world where universal access to computers also meant universal access to information, where housewives could earn graduate degrees from suburban homes and anyone anywhere could trade on the New York Stock Exchange. Rather than consider computing from centralized perspectives, Kemeny saw time-sharing as an explicitly decentralizing force that democratized centers of finance and elite education institutions.[26]

Such a digital utopia was predicated on a far broader penetration of computing. On this front, Dartmouth had reason to be optimistic. In his first annual report on the Kietwit center, Kurtz showed that student use had increased along virtually every dimension: for the year, a total of 64.5% of undergraduate students, including those long past the freshman mathematics class, used the system. Early in 1968, the system reached a peak of 113 simultaneous users, "generally accepted" to be the world record at the time. By the end of the year, the circulation of the Kiewit Comments newsletter reached a thousand subscribers and the Center received visitors weekly from around the world after positive press mentions in publications like *Newsweek* and *Le Monde*. [27]

Not all of the system's users connected from within the Dartmouth campus. In 1940, mathematician George Stibitz, working for Bell Labs, had demonstrated the first remote access of a computer, using a teleprinter in Dartmouth's McNutt Hall to connect to his Complex Number Calculator in New York City. On single-user systems, remote access was more of a novelty. But with time-sharing, access could be shared between a multitude of users, both local and distant. Soon after the installation of the GE-235, Dartmouth had demonstrated the system across the country over long distance telephone lines from San Francisco and, by

---

26 George O'Connell "The Computer Revolution", *Dartmouth Alumni Magazine*, Nov 1964, Kurtz Papers.
"The Computer at Dartmouth", *Dartmouth Alumni Magazine*, Kurtz Papers.
Pamplet, Future Impact of Computers, 1966, box 6, Kurtz Papers.
Future Impact of Computers. Kurtz Papers, Kurtz Papers.

27 "Kiewit Computation Center Fiscal 1967-68 Annual Report to the Trustees of Dartmouth College.", Kurtz Papers.
"Interim Council on Computing Final Report, June 1968. Kurtz Papers", Kurtz Papers.

1967, two dozen colleges and professional schools had experimented with or demonstrated use of the system. Access wasn't limited to post-secondary education. Almost as soon as time-sharing was made available to Dartmouth students in the fall of 1964, Kemeny and Kurtz had a remote terminal installed at the local Hanover High School, "mainly to see what would happen." As with earlier experiments, no curriculum was provided. With little more than the standard BASIC manual, the students proved that youth was no limitation to computing. A group of industrious fifth-graders improved an existing integer factorization program and wrote another to generate magic squares. Others, as young as age 11, worked on projects to plot trigonometric equations, play chess, write poetry, and even compose "moderately good music." By 1967, the school was experimenting with integrating computer use with their own mathematics coursework. Other schools in the region, often subsidized by educational grants from Dartmouth, achieved similarly impressive results; one in Massachusetts trained 400 students (half of their student body) in computer use through a single teletype terminal and planned to institute mandatory computing work for each new ninth grade class. Kemeny and Kurtz believed that, beyond learning to use computers, teaching students to write programs could allow for wholly new ways of learning. They cited the example of a sixth-grader who wrote a program to add fractions. Unlike other programs which coupled computing with more traditional approaches to mathematical education that emphasized the mechanical process of calculation, this student had essentially taught the computer rather than the other way around and thus demonstrated a fundamental understanding of the underlying mathematical concepts.[28]

Remote access to the Dartmouth system was an early form of the commoditization of computing.

---

28 This kind of thinking is in line with Kemeny's advocacy of New Math curricula in primary and secondary education.
Kemeny and Kurtz, "The Dartmouth Time-Sharing Computing System developed under a grant from the Course Content Improvement Program, National Science Foundation, Final Report, June 1967", Kurtz Papers.
"Kiewit Computation Center Fiscal 1967-68 Annual Report to the Trustees of Dartmouth College", Kurtz Papers.
Kiewit Comments, June 19, 1967. 4. Kurtz Papers.

Kemeny and Kurtz wrote of their system as an "Educational Utility" that would level computing's formidable barriers to entry. Now, virtually any college or secondary school could gain access to a multi-million dollar computer for a tiny fraction of that cost, using a simple language that didn't require an extensive framework of engineering and mathematics expertise to learn. By 1967, bolstered by Dartmouth's successes, the Pierce Panel of the President's Scientific Advisory Committee issued a report titled "Computers in Higher Education" that supported the idea of shared regional computing centers for education along the Dartmouth model and two NSF grants were approved to helped fund it. The NSF College Consortium Project granted $35,000 for ten New England colleges to use Dartmouth Time-Sharing, resulting in a thousand new users over six months. A separate and more substantial grant of $142,500 connected a total of 23 regional secondary schools to Dartmouth and helped them coordinate programs and materials for students in grade 7 to 12. Over the course of the 9 month secondary school program, about 5000 secondary students participated, more than the total Dartmouth student population, for a total of 30,000 terminal-hours. As a part of this push, Kurtz experienced an early example of computer networks overloading communications infrastructure. On informing the New England Telephone and Telegraph Company that Dartmouth needed to connect two dozens new schools for eight hours at a time, he learned that his proposal would completely overload the long-distance telephone network for a large region between Boston and Montreal.[29]

With this bottleneck resolved, off-campus usage grew further over the next year. Dartmouth transitioned to the long-awaited and more efficient Phase II time-share software to accommodate the load and formally adopted the acronym DTSS for Dartmouth Time-Sharing System. The College Consortium Project,

---

29 Kemeny and Kurtz estimated that the cost of full-time "normal use" of Time-Sharing through a single teletype terminal was $5000 a year.

Kemeny and Kurtz, "The Dartmouth Time-Sharing Computing System developed under a grant from the Course Content Improvement Program, National Science Foundation, Final Report, June 1967", box 11, Kurtz Papers.

"The Computer in Secondary Education", *Dartmouth Alumni Magazine*, November 1967, Kurtz Papers.

Kemeny unpublished autobiography, Section 6, Box 20, Kemeny Papers. Chapter "Computers", 11.

in its first full year of operation, doubled in size to two thousand users. The second year of the secondary education program added new schools, as far away as Benjamin Franklin High School in Harlem, and an additional thousand users. Some of the earliest participants even installed a second terminal. Such use by secondary schools showed that universal computer education could be established at an even more fundamental level, that it could take root outside the halls of the Ivy League. When George Stibitz had first demonstrated remote access in 1940, the *New York Sun* had reported that no one foresaw a time "when Johnny, aged 12, will be able to pick up his telephone receiver and ask the operator for the answer to seven times nine." In the last years of the 1960s, thousands of students, many as young as 12, were using Dartmouth Time-Sharing over the phone and to solve far more complex problems. Kemeny wasn't too surprised. At *The Future Impact of Computers* conference, he had predicted that the ubiquitous computer access of the future would extend to children: "I do have a daughter in the seventh grade; she knows how to write a good program in BASIC; and she does a lot of her homework on the computer. This is one prediction, therefore, that I make with confidence. There is nothing safer than taking something which is already happening and predicting it will happen 20 years from now."[30]

The Library of the Future: 1970-1976

With each passing year, the number of Dartmouth undergraduate students that ventured into DTSS increased. From just under two-thirds of the total in 1967-1968, Kiewit's first full year of operation, the number steadily grew past 80% in 1971 and to 90% in 1973. Simultaneously, each student was spending more

30 *Sun* piece quoted in "Dartmouth Event of 1940 Made Computer History", *Dartmouth Alumni Magazine*, November 1964, Kurtz Papers.
Byrne, Thomas E. "Kiewit Computation Center Annual Report for the Fiscal Year 1968-1969," September, 1969, Kurtz Papers.
Future Impact of Computers conference, Kurtz Papers, p. 80-83.

time on DTSS on average, more than doubling between 1971 and 1973, and the number of terminals on campus grew, from 11 at the launch of time-sharing on the GE-225 in 1964 to 40 with the installation of the GE-635 in 1966 and 266 by 1973. By the early 1970s, the College was even using DTSS to manage student enrollment and grades. Off campus, the NSF funding for the Regional Computer Consortium ceased in 1970. As a result of their experiences with DTSS, several schools elected to purchase and operate their own computers while the majority remained DTSS users. Despite the end of the NSF program, total off-campus use of DTSS by colleges continued to grow, to over 4000 users in 1973. Other institutions, like the US Naval Academy at Annapolis, established their own clones of DTSS with the aid and blessing of Dartmouth. Use of DTSS by high school students curtailed sharply after reaching a peak of over 6000 in 1971 as funding dried up and schools switched to newly-etablished local computing services or founded their own "mininetworks" with nearby peers. Even in areas where use of DTSS declined, the exposure to computing spearheaded by Dartmouth strongly influenced how those institutions merged computer use and education.[31]

At Dartmouth, development of the DTSS software itself continued to be conducted primarily by undergraduate students, organized into a formal staff of student programming assistants. Students also continued to create their own independent projects. The Computation Center encouraged, but did not mandate a culture of cooperative sharing of these programs through the DTSS Program Library. Students could choose to submit their programs "of general interest" to the library catalog. These would be examined for relevance and correctness and those approved would be added to the catalog, entered into the public domain, and made easily accessible to any DTSS user. By 1974, the Library consisted of hundreds of programs,

---

31 "Kiewit Computation Center Fiscal 1967-68 Annual Report to the Trustees of Dartmouth College." Kurtz Papers. "Kiewit Computation Center Annual Report for the Fiscal Year 1968-1969," September, 1969, Kurtz Papers. "Biennial Report 1969-1971, The Kiewit Computation Center," June, 1971, Kurtz Papers. "Biennial Report 1971-1973, The Kiewit Computation Center," July, 1973, Kurtz Papers. Kemeny, *Man and Computer*, 99-102.

spanning dozens of categories, from language instruction and finance to games and urban planning. The rejection of intellectual property rights to the Library mirrored Kemeny and Kurtz's decision to relinquish rights to BASIC and stood in contrast to the emergence of the commercial software industry; only a few years separated the establishment of the DTSS Program Library and Bill Gates's famous Open Letter to Hobbyists decrying unauthorized copying of software as theft. Dartmouth had long acknowledged the worth of code; the first annual report on the Kiewit Center, writing about the development of DTSS, described software as "an asset of considerable value … equivalent to the cost of the computer equipment." The decision to explicitly pass ownership of the Library to the public domain was a deliberate choice to foster a collaborative intellectual community. The DTSS Program Library represented a tangible manifestation of Kemeny and Kurtz's library analogy for computing, one where students not only frequently and freely checked out works, but also regularly contributed their own back. [32]

Kemeny had also been thinking about more traditional libraries for some time. He had delivered a speech at MIT in 1961 titled "A Library for 2000 A.D." that attempted to address the problem of the growing body of printed works. He estimated that a comprehensive library at the turn of the next century would contain ten million volumes, far in excess of what he felt was reasonable for humans to browse and search and for institutions to maintain. He proposed a National Research Library, funded by the federal government and academic institutions to house these volumes on magnetic tape. Users would search and retrieve desired items through a vaguely defined "projection unit" connected in some fashion to a computer at the library. Although the proposal remained nebulous on technical details, the piece contained hints of time-sharing. Kemeny revisited the idea of the Library ten years later in "Library of the Future", part of a series of lectures he gave at the American Museum of Natural History. He updated his ideas with the technical mechanisms of DTSS; the

32 Kiewit Comments, Vol. 4 No. 2. March 1, 1970, Kurtz Papers.
"Kiewit Computation Center Fiscal 1967-68 Annual Report to the Trustees of Dartmouth College", Kurtz Papers.

new Library would operate regional time-sharing branches to which educational institutions could connect, each of which would contain mirrored collections of all the volumes. By turning the analogy around and modeling the library in the image of the computer, the benefits of time-sharing could be applied to knowledge itself. In the same way that DTSS allowed near-universal open access to computing, the Library of the Future would ensure a similar kind of access to knowledge.[33]

With universal computing a reality at Dartmouth, Kemeny and Kurtz turned their attention to the future directions of the system they had wrought. Kemeny's writings on libraries (and the future he predicted at the *Future Impact of Computers* conference) and Kurtz's policies for the DTSS Program Library envisioned the potent possibilities for information itself, where computers would help humans navigate and access the increasing complex datasets of modern civilization and break down traditional barriers against the free dissemination of knowledge. Tantalizingly, their thinking presages some of the later ideas of Free Software advocates, digital utopians, and crypto-anarchists: that with computers, information could be made free (or something close to it).

Conclusion

Time-sharing wasn't the future. In 1976, the old GE-635 was replaced by a new Honeywell 66/40 after a decade of faithful service and DTSS remained popular with Dartmouth students for a time. For years, Kiewit Comments could still claim that each incoming cohort would make use of time-sharing at rates up to 95%. But 1977 would see the release of the first truly successful personal computers: the Commodore PET, Apple II, and the TRS-80. All three ran variations of the BASIC language adapted for use on microcomputers. Two years later, Kiewit acquired Apple and Terak microcomputers for students and faculty to experiment on. In the fall

33 Kemeny, *Random Essays on Mathematics, Education, and Computers*, 134-156.
Kemeny, *Man and the Computer,* 72-98.

of 1984, Dartmouth began encouraging new students to purchase the Apple Macintosh announced in the 1984 ad, available as part of a standard Dartmouth package for $1265. The reasons why personal computing beat out time-sharing, within the academy and without, are many: advances in computer hardware upended the cost benefits of time-sharing; increasingly, the interfaces of computer systems distinguished between programming and use; centralized computing, no matter how liberally run, necessarily mandated a degree of inflexible control. At the launch of the Kiewit center in 1966, Donald Kreider had credited "a certain amount of mischief" with the success of Dartmouth's early computing efforts; a factor that distinguished the Dartmouth system from any variety of microcomputer was simply that the latitude for mischief on an university mainframe was necessarily less than on a personal machine. Whatever the reasons, at Dartmouth and elsewhere, time-sharing eventually faded away.[34]

Regardless of the ultimate obsolescence of time-sharing, computing at Dartmouth aptly demonstrated the ample liberal possibilities of mainframe systems. Universal computer access, the leveling of student-professor hierarchies, and the fostering of the free exchange of information, much more so than DTSS or BASIC were its truly revolutionary products. In many ways, Dartmouth represented a unique confluence of the right institutional culture, time, and people for these possibilities to flower. The college was a small but prestigious institution with a strong mandate dedicated explicitly to undergraduate liberal arts education rather than research or professional training. The 1950s and 1960s was a period of great anxiety and re-evaluation, but also government re-investment in American technical education and simultaneously a time of tremendous technological advancement. A decade earlier, open access college computing would have been impossible technically and financially. Kemeny was a brilliant and charismatic young professor given carte blanche to reform Dartmouth's mathematics department and Kurtz a capable administrator who shared his vision. Behind

34 Kiewit Comments, Vol. 13 No. 2, Winter 1979, Kurtz Papers.
MicroComputer Notes from Kiewit, Vol. 1 No. 1, February 20, 1984, Kurtz Papers.

them was a generation of capable young hackers who repeatedly proved the merits of the virtually unqualified freedoms they were granted.

The potent combination of all these factors made Dartmouth an exceptional case amongst a cohort of heavyweight research universities tied to government research and defense funding and with formalized programs designed to meet that need. Thayer School of Engineering dean Myron Tribus once recounted a conversation he had at a national conference with a peer from such an institution, who told Tribus of the myriad of formal computer coursework that supported his program. Tribus asked: "We are a very small school and I don't think we could support that many courses. Do you really need them?"

"Well, there are some small schools that will probably not be able to make contributions to the field of computation. Where are you from?"

"I am from Dartmouth."

"Oh! I'm sorry, I didn't mean *you*!"[35]

35 Myron Tribus, Future Impact of Computers conference, Kurtz Papers, 18-25.